

Consultas multimesa.

Contenido

Introducción	1
Cómo se hacía en versiones anteriores. Composiciones Internas	2
Alias de tabla	4
Composiciones en la versión SQL99.....	5

Introducción

Una de las propiedades de las bases de datos relacionales era que distribuíamos la información en varias tablas que a su vez estaban relacionadas por algún campo común (Clau Ajerna). Así evitábamos repetir datos. Por tanto, también será frecuente que tengamos que consultar datos que se encuentran distribuidos por diferentes tablas.

En versiones de SQL anteriores a 1999, la vinculación de tablas se realizaba en la cláusula WHERE. Por la versión SQL de 1999 especifica una nueva sintaxis para consultar varias tablas. La razón de esta nueva sintaxis era separar las condiciones de asociación respecto a las condiciones de selección de registros (files) y la asociación se realiza en la cláusula FROM.

Aunque SQL admitiría cualquier criterio compatible (que los campos de las dos tablas tuvieran dominios compatibles) para el enlace de tablas. Por ejemplo: si el DNI de una persona es un CHAR(9) y el código de un producto de Carrefour también es un CHAR(9), podríamos enlazar las dos tablas de forma que consultamos qué personas tienen un DNI que coincida con alguno código de un producto de Carrefour.

Como notará, es muy improbable que nos hagamos esta pregunta. Las consultas suelen tener una lógica de modo que se enlaza la Llave Ajena de una mesa con la Llave Primaria de la mesa referenciada por aquélla.

Debemos saber en todo momento qué es lo que estamos obteniendo al enlazar las tablas:

Cuando vinculamos la clave externa de la tabla A con la clave primaria de la tabla B (a la que hace referencia A), normalmente, queremos obtener las filas de la tabla A, que tienen valor en la Clave Aliena, con las filas de B correspondientes. Es como si tuviéramos las filas de la tabla A con los datos correspondientes en la mesa B ampliadas.

Cómo se hacía en versiones anteriores. Composiciones Internas

Como se ha dicho, en primeras versiones de SQL el enlace de tablas se hacía en la cláusula WHERE como una condición más, normalmente, de obligado cumplimiento para todas las filas (AND (resto de condiciones)). Ésta era la única opción de enlazar las tablas en las primeras versiones de SQL.

Nosotros no lo haremos así, pero es interesante verlo.

¿Qué ocurre si combinamos dos o más tablas sin restricción alguna? El resultado será un producto cartesiano. El producto cartesiano entre dos tablas da como resultado todas las combinaciones de todas las filas de esas dos tablas. Se indica poniendo en la cláusula FROM las tablas que queremos componer separadas por comas. Y puedes obtener el producto cartesiano de las tablas que quieras. Como lo que se obtiene son todas las posibles combinaciones de filas, debes tener especial cuenta con las tablas que combinas. Si tienes dos tablas de 10 filas cada una, el resultado tendrá 100 filas (10 x 10), a medida que aumentamos el número de filas que contienen las tablas, mayor será el resultado final, con lo que puede considerarse que nos encontraremos con una operación costosa.

Esta operación no es de las más utilizadas ya que coge una fila de una mesa y la asocia con todos y cada uno de las filas de la otra tabla, independientemente de que tengan relación o no. Lo más normal es que queramos seleccionar los registros según algún criterio. Necesitaremos discriminar de alguna forma para que únicamente aparezcan filas de una tabla que estén relacionadas con la otra mesa. A esto se le llama asociar o enlazar tablas (JOIN). Para hacer una composición interna se parte de un producto cartesiano y se eliminan aquellas filas que no cumplen la condición de composición. Lo importante en las composiciones internas es emparejar los campos que deben tener valores iguales. Las reglas para las composiciones son:

En la cláusula SELECT, se pueden citar columnas de ambas tablas, condicionadas o no, la combinación.

Las columnas que aparecen en la cláusula WHERE se denominan columnas de apareamiento ya que son las que permiten emparejar las filas de las dos tablas. Éstas no tienen por qué estar incluidas en la lista de selección. Emparejaremos tablas que estén relacionadas entre sí, clave ajena de una mesa con la clave primaria de otra. Cuando emparejamos campos debemos especificar de la siguiente forma:

```
...  
WHERE NombreTabla1. Camporelacionado1 =  
NombreTabla2.Camporelacionado2.
```

Ejemplo:

Trabajaremos en la BD 'Jardinería' donde disponemos de una mesa OFICINAS (cuya clave principal es CodigoOficina) y esta tabla a su vez está referenciada por la tabla EMPLEADOS por el campo CodigoOficina de la mesa EMPLEADOS que tiene restricción de Clau Aliena hacia OFICINAS. Si quisimos obtener los datos de los empleados incluyendo los datos completos de las oficinas en las que trabajan, necesitaríamos datos de ambas tablas.

Supongamos este estado de las tablas EMPLEADOS y OFICINAS:

TALLERES			EMPLEADOS		
CódigoOficina País		Ciudad	CodigoEmpleado Nombre	CodigoOficina	
VA-ES	España	Valencia	1	Bien	VA-ES
MA-ES	España	Madrid	2	Alberto	VA-ES
PA-FR	Francia	París	3	Javier	PA-FR
	Italia	Milán	4	María	PA-FR
			5	Sofía	PA-FR

Al hacer el producto cartesianos de las dos tablas obtenemos:

Seleccionar * From EMPLEADOS, OFICINAS					
Camps de EMPLEADOS			Camps de OFICINAS		
CodigoEmpleado Nombre		CódigoOficina	CódigoOficina País		Ciudad
1	Bien	VA-ES	VA-ES	España	Valencia
1	Bien	VA-ES	MA-ES	España	Madrid
1	Bien	VA-ES	PA-FR	Francia	París
1	Bien	VA-ES		Italia	Milán
2	Alberto	VA-ES	VA-ES	España	Valencia
2	Alberto	VA-ES	MA-ES	España	Madrid
2	Alberto	VA-ES	PA-FR	Francia	París
2	Alberto	VA-ES		Italia	Milán
3	Javier	PA-FR	VA-ES	España	Valencia
3	Javier	PA-FR	MA-ES	España	Madrid
3	Javier	PA-FR	PA-FR	Francia	París
3	Javier	PA-FR		Italia	Milán
4	María	PA-FR	VA-ES	España	Valencia
4	María	PA-FR	MA-ES	España	Madrid
4	María	PA-FR	PA-FR	Francia	París
4	María	PA-FR		Italia	Milán
5	Sofía	PA-FR	VA-ES	España	Valencia

5	Sofía	PA-FR	MA-ES	España	Madrid
5	Sofía	PA-FR	PA-FR	Francia	París
5	Sofía	PA-FR		Italia	Milán

Como observarás, para nuestra consulta sólo tienen sentido lógico las filas sombreadas. Las que enlazan a cada empleado con su oficina

<p>Seleccionar *</p> <p>From EMPLEADOS, OFICINAS</p> <p>WHERE EMPLEADOS.CodigoOficina=OFICINAS.CodigoOficina</p>					
Campos de EMPLEADOS			Campos de OFICINAS		
CodigoEmpleado	Nombre	CódigoOficina	CódigoOficina	País	Ciudad
1	Bien	VA-ES	VA-ES	España	Valencia
2	Alberto	VA-ES	VA-ES	España	Valencia
3	Javier	PA-FR	PA-FR	Francia	París
4	María	PA-FR	PA-FR	Francia	París
5	Sofía	PA-FR	PA-FR	Francia	París

Alias de tabla

Cuando en el FROM citamos una tabla es posible seguidamente darle un sobrenombre que podremos utilizar en cualquier parte de la consulta.

Como después veremos, los sobrenombres de mesa son a veces imprescindibles. La sintaxis sería:

```
DE Tabla1 Alias1...
```

Puedes combinar una mesa consigo misma, pero debes poner de forma obligatoria alias las tablas que repetirás. Veamos un ejemplo, si queremos obtener el nombre de un empleado y el nombre de su jefe, debemos enlazar los datos del empleado con los datos de su jefe. Es decir, necesitamos dos tablas EMPLEADOS: una jugará el papel del empleado y otra el papel de cabeza. Tendremos:

```
SELECCIONE Nombre.EMP, Nombre.JEF
FROM EMPLEATS EMP, EMPLEADOS JEF
WHERE EMP.CodigoJefe=JEF.CodigoEmpleado;
```

Se puede prescindir del nombre de la tabla si el nombre del campo no se repite en ambas tablas y resulta imprescindible hacer referencia a la misma (o a su alias) si no es así.

Composiciones en la versión SQL99.

Nosotros enlazaremos las tablas así: en la cláusula FROM.

Como has visto, SQL incluye en esta versión mejoras de la sintaxis a la hora de crear composiciones en consultas. La sintaxis que utilizaremos es la siguiente:

```
SELECCIONAR tabla1.columna1, tabla1.columna2, ..., tabla2.columna1,
tabla2.columna2, ...
DESDE tabla1 {UNIÓN NATURAL tabla2 | UNIÓN INTERNA tabla2 ON
tabla1.columna=tabla2.columna | JOIN tabla2 ON
tabla1.columna=tabla2.columna | ÚNETE A LA IZQUIERDA
tabla1.columna=tabla2.columna | ÚNETE A LA DERECHA
tabla1.columna=tabla2.columna | UNIÓN EXTERNA COMPLETA tabla2 ON
tabla1.columna=tabla2.columna};
```

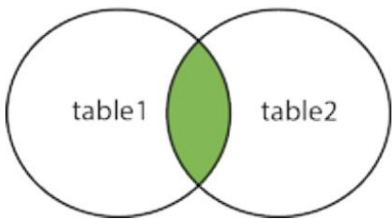
- NATURAL JOIN: detecta automáticamente las claves de unión, basándose en el nombre de la columna que coincide en ambas tablas. Por supuesto, se requerirá que las columnas de unión tengan el mismo nombre en cada mesa.

funcionará incluso si no están definidas las claves primarias o ajenas (enlaza por campos del mismo nombre en las dos tablas) Hay que tener cuidado de que SÓLO compartan nombre los campos que nos interesa enlazar. Si no es así (hay otros por ejemplo), es necesario utilizar UNIÓN INTERNA.

Seleccionar * From EMPLEADOS NATURAL JOIN OFICINAS				
Campo Común	Camps de EMPLEADOS		Camps de OFICINAS	
CodigoOficina	CodigoEmpleado	Nombre	País	Ciudad
VA-ES	1	Bien	España	Valencia
VA-ES	2	Alberto	España	Valencia
PA-FR	3	Javier	Francia	París
PA-FR	4	María	Francia	París
PA-FR	5	Sofía	Francia	París

- JOIN ON | INNER JOIN ON: se utiliza para unir tablas en la que los nombres de columna no coinciden en ambas tablas o se necesita establecer asociaciones más complicadas.

INNER JOIN



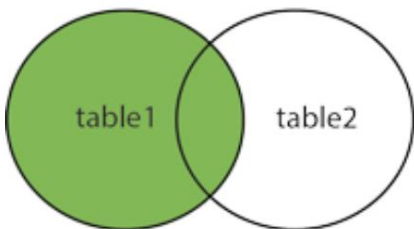
Seleccionar *					
From EMPLEADOS INNER JOIN OFICINAS ON					
EMPLEADOS.CodigoOficina=OFICINAS.CodigoOficina					
Camps de EMPLEADOS			Camps de OFICINAS		
CodigoEmpleado	Nombre	CódigoOficina	CódigoOficina	País	Ciudad
1	Bien	VA-ES	VA-ES	España	Valencia
2	Alberto	VA-ES	VA-ES	España	Valencia
3	Javier	PA-FR	PA-FR	Francia	París
4	María	PA-FR	PA-FR	Francia	París
5	Sofía	PA-FR	PA-FR	Francia	París

- HOJA OUTER JOIN ON: muestra todas las filas de ambas tablas. Las que no enlazan tendrán los campos correspondientes a la otra mesa a null. (no la gastaremos)

Seleccionar *					
From EMPLEADOS FULL OUTER JOIN OFICINAS ON					
EMPLEADOS.CodigoOficina=OFICINAS.CodigoOficina					
Camps de EMPLEADOS			Camps de OFICINAS		
CodigoEmpleado	Nombre	CódigoOficina	CódigoOficina	País	Ciudad
NULO	NULO	NULO	MA-ES	España	Madrid
NULO	NULO	NULO		Italia	Milán

- LEFT JOIN ON: es una composición externa izquierda, todas las filas de la tabla de la izquierda se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas. En nuestro caso, ya estaban todas las filas de EMPLEADOS

LEFT JOIN



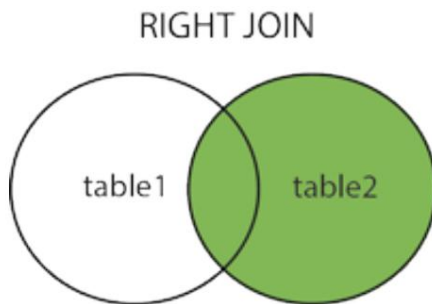
Seleccionar *

From EMPLEADOS LEFT JOIN OFICINAS ON

EMPLEADOS.CodigoOficina=OFICINAS.CodigoOficina

Camps de EMPLEADOS			Camps de OFICINAS		
CodigoEmpleado	Nombre	CódigoOficina	CódigoOficina	País	Ciudad
1	Bien	VA-ES	VA-ES	España	Valencia
2	Alberto	VA-ES	VA-ES	España	Valencia
3	Javier	PA-FR	PA-FR	Francia	París
4	María	PA-FR	PA-FR	Francia	París
5	Sofía	PA-FR	PA-FR	Francia	París

- RIGTH JOIN ON: es una composición externa derecha, todas las filas de la mesa de la derecha se devuelven, aunque no haya ninguna columna correspondiente en las tablas combinadas.



Seleccionar *					
From EMPLEADOS RIGHT JOIN OFICINAS ON					
EMPLEADOS.CodigoOficina=OFICINAS.CodigoOficina					
Camps de EMPLEADOS			Camps de OFICINAS		
CodigoEmpleado	Nombre	CódigoOficina	CódigoOficina	País	Ciudad
1	Bien	VA-ES	VA-ES	España	Valencia
2	Alberto	VA-ES	VA-ES	España	Valencia
3	Javier	PA-FR	PA-FR	Francia	París
4	María	PA-FR	PA-FR	Francia	París
5	Sofía	PA-FR	PA-FR	Francia	París
NULO	NULO	NULO	MA-ES	España	Madrid
NULO	NULO	NULO		Italia	Milán

Aquí tienes una guía visual del resultado obtenido con los diferentes joins y condiciones:

